

# ギークとウェブの方向性

The direction of geeks and web

水野 貴明

株式会社はてなにて勤務しております水野と申します。今日僕は「ギークとウェブの方向性」ということで、早速お話のほう始めさせていただきますと思います。

まずウェブ上で展開されるユニークな色々なサイトのサービスが出てきますが、その多くはギークと呼ばれている人達によって作られています。本日はそのギークというのは何かということから始まって、ギークと言われる人達の行動や考え方からウェブのサービスというのが今後どうなっていくのかについて考察していければなという風に思っております。

ギークという言葉いきなり何の説明もなく使いましたが、まずギークは何かということ、何故注目されているのか、どういう性格をしているのか、そしてギークがウェブサービスの進化とどう関係しているのか、最後に、時間に余裕がありましたらギークがどういう生活を送っているのかといったことも、潜入取材がございますので、その辺も見えていただければと思っています。

私は「株式会社はてな」という所でエンジニアとかプログラマーとかいうか、開発業務に携わっております。年齢は33歳、昭和48年生まれでございます。「はてな」の中のエンジニアの中では最年長でございます。大変年を重ねてしましまして、若者はだいたい24ぐらいからいて、そういうちょっと若い会社です。社長も僕より年下です。それ以外に

テクニカルライターということで、雑誌に本を書いたり、雑誌に文章を書いたり本を書いたり、文章を書く仕事もしております。本とかを年間に5冊ぐらい書いたりとかして、「お前、いつ仕事してるんだ」とか「いつ寝てるの」とか言われているんですけど、主に技術系の、プログラミングの本から先程も滝日さんの話にもありました、RSSのちょっと技術よりの話の本とか、あとはGoogleとかの使い方みたいなガイドの本まで色々取り揃えて書いていますので、本屋さんでお見かけした際にはぜひよろしくお願いいたします。

「株式会社はてな」は何の会社かについて、前提となる話ですので、簡単に説明させていただきます。インターネット上での色々なウェブサービスサイトを手がけているベンチャーです。インターネット上の情報活用、情報収集とか、そういったものに役立つサービスを色々展開しています。滝日さんのお話にもありました「はてなブックマーク」も「はてな」のサービスの1つです。他にもRSSリーダー、ブログサービス、それから人力検索、これらが柱です。Googleとかですと、ロボットプログラムが検索結果を返してくるわけですけども、それだと分からないような質問を投げ掛けておくとそれが公開されて、誰か読んだ人が答えてくれるみたいなサービスを行っております。拠点のオフィスが東京の渋谷のはずれにありまして、社員21人です。去年の9月に入社してちょうど1年なのですけれども、僕が入る前までは社員10人

で、1年間で倍になりました。7月にアメリカに「はてなインク」という会社があって、社長がアメリカに行ってしまいました。アメリカというか英語圏というか、世界に向けてサービスをやっていきたいと言っている状況にある会社です。こんな青っぽい、白に青基調という色で統一したサイトを色々やっているの、こんな色のサイトを見かけたら「はてな」かもなと思っていただければ幸いです。

今回なんで「はてな」の話を最初にしたかという、社員の半分以上が技術者で、技術者の割合ってというのが去年からほとんど変わってないけど社員の数が2倍になっているので、かなり技術者が幅を利かせている会社なのです。仕事は平日に勿論やっているわけですが、休みの日も何か家でプログラム書いていたり、プログラムを書くのが大好きな人が多い会社です。技術者が大きな権限を持っているというのは、社内でどういうことがやりたいとか、何を作っていくかということを技術者が決めている、技術者が大きな力を持っていて、発言権とか決定権とかを持ってやっている会社です。社長も技術者というのもあるのですが、社長もプログラムを書いている、そういう会社です。

とりあえずコードを書くことが重要で、コードっていうのはプログラムと同じ意味なのですけれども、プログラムを書くのがとにかく好きで、プログラムを書いてないと技術者として全然認められない会社なのです。実はそれというのがこれからお話する内容ギークという話とすごく関わってきます。

ギークっていうのは geek と書きます。これは英語です。辞書で引くと、「変人、奇人、変態、異常者、ろくでなし、バカ、アホ、マヌケ、グサイ男、酔っぱらい、飲んだくれ」とあります。こういう人達がウェブサービス作っているというのは、あながち間違いでもないかも知れないのですが、元々の意味はこういった意味です。はっきり言ってしまう

とネガティブな、いい言葉ではないという感じなのですが、更に「低俗、生きた鶏などの首を食いちぎる異常な見せ物師」と goo 辞書には書いてありまして、なんじゃこりゃという感じなのです。元々はこういうサーカスなんかの見せ物をやる人達のことをギークと呼んでいた経緯があるのですが、それが最近ちょっと意味が変わってきました。いい意味で使われるようになってきたのです。どんな意味かというと「ウェブの世界で最先端技術をいち早く取り入れ、ITに精通するエンジニア」という褒め言葉としてのコンピューターおたくということです。先程「はてな」という会社は、経営者が技術者で技術者が力を持っていて、技術者の権力が高いという話をしましたけれど、みんなすごくコンピューターが大好きということです。コンピューターおたくって言ってしまうと悪い意味、ネガティブな意味も含んでいますけれども、「あいつはすごいコンピューターに詳しくて、コンピューターのことが無茶苦茶好きで、コンピューターと一緒にいればメシもいらんんじゃないか」みたいな感じの、コンピューターが大好きでそれを上手く活用出来ている人間のことをギークと呼んでいます。ということで、「はてな」にも割かしギークがいるのではないかなと思っています。もう1つは「ウェブの世界で最先端技術をいち早く取り入れ、ITに精通するエンジニア」と書いてありますが、本当にその世界は、情報の流れが速いですし、技術もどんどん進歩して、ドッグイヤーなんて、1年が7年にもこの世界では相当するなんてことも言われますけれども、そういった感じでどんどん進歩していったわけなので、それを取り入れて職業としてやっていくというのはハードで、好きじゃないと出来ないことなのですから、そういうのを追っかけるのが大好きで、いつも新しいこと何か出来ないかなとか、こういうことやったら面白いんじゃないかなとかを

考えてどんどん新しいことを開拓していくような、ちょっとフロンティアスピリット溢れるとちょっとカッコ良すぎなのですけども、頑張っているエンジニアのことを一応ギークと、褒め言葉として使うのです。だからみんなで自分のことをギークだっていうのは、僕は色々そのギークな人達と最近非常によく付き合うようになって、自分がギークかどうかちょっと、ギークじゃないんじゃないかなって思うようになってきたんですけども、あの人はギークだっていうのは、「すごいあいつはすごい技術を持っている」と同じ意味だと思っていただければと思います。

類義語としてのハッカーというのは悪い意味でも使われちゃったりしていますが、本来は非常にプログラミングに精通してすごい技術を持っている技術者というか、プログラマーのことをハッカーというわけです。更にウィザードとかグルというのがあり、ウィザードは魔法使いで、グルは導師という言葉から来ていますけど、ウィザードとかグルっていうのはハッカーよりすごい人達のこと元々いう言葉らしくて、Linux という OS、オープンソースで作られている有名な OS がありますけれども、あれを作ったリーナス・トーバルズという人は Linux のグルだと言われています。導く師と書いて導師です。ナードというのは、ちょっとネガティブな言葉としてのおたくっていう意味らしいのですけど、これもまあ近いって言えば近いかなという感じです。こんな類義語がある言葉としてのギークという言葉なのですけども、それを押さえていただければと思います。

ギークの対比としてスーツという言葉があるのです。スーツというのは、いわゆる会社、会社組織の中での話なのですけど、営業をやっている人です。普段スーツ着ている人達のことです。インターネットもすごい大きなビジネスになってきていますので、どうしてもやっぱり営業さんですとか、意志決定をす

るそういった人っていうのは重要になってくるのですけど、やっぱり考え方っていうのがスーツと呼ばれる人達とギークと呼ばれる人達というのは、方向性が違うのです。なので、スーツとギークという対立としての言葉が生まれて、お互いに「あいつは何も分かってない」ということを言い合う仲、勿論そういう人達ばかりじゃないですけども、ステレオ的に言えば、そういった対比が生まれ、スーツに対してギークっていう違う方向性を向いているということです。会社とかっていうのはお金を儲けることが大事で当たり前なのですけども、スーツの人達がいかに自分の会社の価値を高めていくとかか、どうやって売っていくかということを中心に考えたりするのですが、一方でギークはもう技術指向で、「そんなのは技術的に美しくない」とか言ってそういうのを否定してみたりと、だいふ考え方が違うのです。ただ最初のテーマにもあるようにギークと呼ばれている人達の動きというのが、最近インターネットの中では重要になってきているのでないかなということ

です。じゃあギークはプログラマーのことなのかとなるのですけど、日本語でプログラマーというイメージが誤解を招きやすいのです。旧来からシステム開発のイメージだと、プログラマーというのは、上から階層になっていて、一番上によく分からない社長とか偉い人がいて、その下にプロジェクトマネージャーとかいうプロジェクトを統括する人がいて、下にシステムエンジニアという、いわゆる上流工程というシステムの設計をしたりデータベースの構成考えたりというプログラムを書くのではなく、プログラムの設計をする人達がいて、そういう人達が作った物をプログラマーがコーディングするという、プログラムを書くだけみたいな、システムというか会社システムの中で働いている人とかもいっぱいいるのですけど、いまだにそういう感じがプ

プログラマーというのだという位置付けになります。別名 IT 土方とか言われて、本当に IT 産業がすごく流行っていてプログラマーの求人も多いって言うけど、入ってみると言われたプログラムをただひたすら書いただけみたいなそういった現象も生まれてきてまして、そういった人達のイメージがプログラマーという言葉には、日本では、強いと思います。ギークっていうのはそうではなくて、自分の意志を持って自分がやりたいことをやっている、好きでプログラムを書いている人達のことを指す言葉だと思います。

ティム・オライリーという人がいまして、このティム・オライリーがやっているオライリーメディアという会社があり、出版だけでなく、各種カンファレンスの主催ですとかいったことも、手広く広げてやっています。ティム・オライリーは、とにかくエンジニアとかハッカー、ギークと言われている人達の動向をずっと見守っている人なのです。毎年このオライリーがオライリーレーダーで技術がどういう風になっているのかということを書きノートとして最初に総括みたいなことを言うのです。Web 2.0 という言葉は、2005 年にこのオライリーレーダーで喋ったと思うのです。

1 年前の 2004 年のオープンソースコンベンション (OSC) での話の中で、アルファギークという言葉が出てきました。アルファギークというのは、「先鋭的なハッカー、とがったエンジニア、最先端のプログラマー」、アルファというのは第一のとか先端のとかっていう、まあアルファ、ベータ、ガンマのアルファですけど、トップを走るみたいな意味がある言葉としてこの後流行っています。有名なブログを書いている人達をアルファブロガーと言ってみたりと、色んな言葉も発生したのです。アルファギークというのはそういう風にとがっていて、トップランナーとして走っているプログラマー、ハッカー、エンジニアで、

革新的なテクノロジーを見つけてきて、それをとにかくいりじまわす、そうすると色んな可能性というのが見えてきて、アルファギーク達がどうしたいかということではなくて、その技術自体がいったいどういう方向に向かっているのかということが、見えてくる、そういう役割を持っている人達のことをアルファギークと呼ぶのだとティム・オライリーは 2004 年に言いました。それはこれから話すギークってものの話にすごく合った話なので紹介したのですが、アルファギークと言われる人達が最新鋭の技術をどんどん使って色んなものを生み出していくので、こういう人達も重要な役割をウェブの世界で担っています。僕が見たことを付け加えると、好奇心が旺盛で新しいもの大好き、更に飽きっぽい。ティム・オライリーも、「彼等はすごい最先端のものをどんどんいじり倒すけど、それがようやく周りの人達が認め出すと、もうその頃には飽きて新しいものに行っちゃってるんだ」と。実際アルファギークと言われている人達、ギークと言われている人達は飽きっぽい性格があるなと思っています。好奇心が旺盛で飽きっぽくてプログラミングがすごく大好き、コードを書いたり、コンピューターの前に座って何かするのも大好き、そういうことを考えると、昔の発明家、エジソンとかあの辺の人達に近い存在なのかなと感じています。

更に加えて、情報収集能力が高い。新しい技術というのはたくさん情報があるので、それを取捨選択して必要な物を取り入れていく能力が問われる人達なのですけども、そういうのが上手い人が、アルファギークと呼ばれているのかなと思います。とにかく自分が作ったものに自信を持っている人達が多くて、そういう人達が出てくるこういうカンファレンスに行くと「こんな俺すごい物作ったよ」みたいなことをみんな口々に言っていて、どうしちゃったのだらうと思うのです。

すごいプライドが高くて、自信過剰です。更に批判に弱いです。すごく頑張って作ったものを、ネットでこんな全然ダメって言われるとすぐシオシオってなっちゃって、出したものをすぐ引っ込めちゃったりして、プライドが高い割に褒めて欲しい人達なのだな、褒めて育つタイプなのだと思います。

ギーク同士は技術力を、どんなコードを書いたのか、どんなものを公開したかで、評価するということです。ギークの人達ってというのはプログラミング、相手がどういうプログラムを書くか、相手がどんなことを知っているかで相手を評価します。なので、外国のギークのカンファレンス、たとえばオープンソースコンベンションに行くと、日本から来たというと、「へー、そうなんだ」みたいな感じで、「どういう会社にいるの」って聞くから「ブログとかやってる会社なんだよ」「そうなんだ、ふーん」という感じなのですが、こういうコードを書いてこういうところで公開してまですって話になったりすると、「それってどういうコード？」みたいな、突然態度激変ということが起こるということです。ギークの人達はプログラムをどう書いているか、どういうものを書いたかが相手を見る基準にしています。彼等はいつもコードを書いている、とりあえず分からないからプログラムを書いてみる、色々御託を並べるのだったらまずコードで示せという感じの勢いでやっている人達です。

ギークに何故注目しなきゃいけないのかということをお話したいと思います。色んなサービスが今登場しています。Google, YouTubeが最近Googleに買われて、大変話題になりました、flickrは写真共有サービスで、デリシャスまたはてなブックマーク、ソーシャルブックマークサービスでdel, icio, Zohoというのはウェブ上でオフィスが使える、ディグっていうのはみんなが注目しているニュースがよく分かるというニュースサイトです。

これらの革新的なサービスの多くはギークと呼ばれる人達が作っているということが、作った人とかの話とかを聞くと思われるのです。

ギークの人達が作ったサイトがこんなに注目されているという話をすると、まず前提として場が整ったというのがあるのです。「インフラの整備と低価格化」ということで、誰でもサービスを提供出来るようになったという現状があります。安くインターネットに高速で接続できてサーバー立てられるようになったのです。それから「オープンソースソフトウェアの開発と発達」ということです。OSだったらLinux、データベースソフトだったらMySQLとか、ウェブサーバーとかそういったソフトがほとんどオープンソースで使える。無償で使える状態のものがたくさん出てきたのです。一昔前ですと大規模なサーバーとかで、大規模なサイトやろうとするとどこかの企業のでかいサーバーを入れて、そのOSを入れて、その提供しているサーバーを入れて、メンテ要員とかも年間契約を結んでとすごくお金がかかったのですけれども、最近そんなのなくても作れるようになりました。オープンなデータというのは情報が出るのですが、最近は色んなデータが無償で公開され使えるようになってきて、割とその自由に色んな情報が加工出来るようになったというのがあります。そういう感じなので、あとはやる気とアイデアがあって知識をがっちり身に付ければ、どんどんサービスを作って公開出来るようになりました。1人でも2人でも、サービスが作れば、大規模な会社とかでなくても簡単にインターネット上でサービスを始めることができるようになって、立場と機会がみんなに平等に訪れるようになった結果、きちんと能力がある人が目立つようになった。その結果色んなサービスが生まれるようになってきたというのがあります。例えばディグという注目されている

ニュースが上に上がってくるサイトというのは、最初のサービスを作るのに 20 時間でプログラムできたみたいなのです。月 99 ドルのサーバーを借りて、20 時間ぐらいで作ったソフト、プログラムでスタートができ、今すごく有名になっているサイトですけれども、ごく簡単に始められているのです。簡単で誰でもチャレンジできて、結果面白いものが色々出てくるようになっています。

ギークがどういう考えに基づいてサービスを作っているか、どういうことを思っているのっていうことを考えると、次世代のウェブサービスというのはこういう人達の考えていることを見ると、この道筋は正しいのか、この道筋は間違っているのかどうか、こうなりそうとかいうことが分かるのではないかなということで、この後を続けていくという話になります。

ギークとウェブの発展ということで、ウェブをギークがどういう形で考えていたから今のウェブが発展したのかという話にしたいと思います。Web 2.0 という言葉が流行っておりますが、私はこの講演においてほとんどこの言葉は使いません。何故かという、概念が広すぎて今回の主旨にそぐわないからです。今回私は内容を絞り込んで喋りたいということもあり、それから Web 2.0 という言葉は 2005 年にオリイが言ったのですが、「ウェブが最近すごく進化してきたよね、だから現状は 2.0 って言えるよね」とその時点でのスナップショットに過ぎないと思っているからです。Web 2.0 が今後も続いていくというわけではなくて、Web 2.0 が 2005 年のことで多分今は何か違う何かじゃないかなという風に思っているところなのです。それから 3 番は飽きたってことです。最近たくさん使われているので少し飽き気味です。ちょっと食傷気味じゃないかなと皆さん思っているのではないかと思っているのです。もう Web 2.0 とかいったことじゃなくて、どうい

うことができてどういうことが面白いのかという切り口で考えようという感じになってきています。Web 2.0 という言葉は使わなくなりました。スナップショットという意味で、1.0, 2.0, 未来というのがあって、ウェブサイトというのは色々な形で毎日ギークの人達を始め、試行錯誤してこんなの面白いのではないかと、こんなのがいいのではないとか、淘汰が起こったりしているわけですが、ウェブサイトの進化をソフトウェアの進化になぞらえたら、今 2.0 ぐらいじゃねーの、1.0 だった過去からちょっと一皮むけたよねって去年の話で 2.0 という名が付いたので、今こっというスナップショットだなという話です。概念としての 2.0 とは、オリイが言ったことをよくよく考えてみるとこういうのがウェブで重要なんじゃないか、ウェブサービスが色々試行錯誤されたことによって、こういうことが流行であるということが分かってきたということいくつかあげられていきます。それは、双方向性、インタラクションという、単に情報を配信するだけでなく情報を共有したり、ユーザーがコンテンツを当てていたりとかっていうことで起こる何かです。集合知と言われていて、『みんなの意見』は案外正しい』という書籍も出ていますが、いろいろな人何千人といた意見をどんどん集めていくと、1人や2人の識者がこれだって言ったものよりも、結果として正しい知見が得られるといった調査結果が出たりとかして、ウェブでたくさんの人が繋がっているわけですから、情報を集めることでより人間は賢くなれるのではないかという話が集合知です。あとフォークソノミーに関しては滝日さんの話にも出てきましたけれども、タクノソミーという分類学という言葉とフォルクスという民族とか民衆という言葉の合成語で、よってたかって人々が分類をするタギングといわれている技術が生まれてきました。ロングテールっていうのは、

アマゾンに代表されるように、在庫を持たずにインターネット上でたくさんものが売れるので、ほとんど売れないものでもそれを集めると意外な額になっていく、グラフで描くとほとんど売れない、売れ筋商品から遠い部分、長いシッポの部分が意外と重要なのではないかなってきいたのがロングテールです。こういったものが有効だってことは、いろんな見地から言われていて、有効だと思います。なんでこういったものが生まれてきたのかってことを考えてみると、ギークの特性というかものが出てきたなという風に私は考えています。プログラマー三大美德という言葉があります。Perl という非常に有名なプログラミング言語がありますが、ラリー・ウォールという日本が大好きでアニメが大好きなおじさんが作ったのです。「プログラミング Perl」というオライリーから出ている技術書があって、これは、Perl という言語を作った人が書いた經典みたいなものなのです。ラクダの絵が描いてあるのでラクダ本と言われます。この中でプログラマーには3つの美德「怠惰」「短気」「傲慢」があると言っています。美德？っていうと、ちょっと待ってみたい感じがするのですが、この3つがあるおかげでプログラムは素晴らしい、これはプログラマーの持つもっともいいところだと書いています。それだけじゃ意味が分からないので、この特性が非常に Web 2.0 的な考え方というか、新しいウェブの考え方に影響を与えていると思います。

まず「怠惰」、laziness というのが何かというと、同じことを何度も繰り返すのはものすごく嫌いという考え方です。とにかく、例えばこうプログラムだから、プログラムを書いてコンピューターにやらせてしまえ、コンピューターというのは単純作業とか繰り返し作業というのは得意ですから、コンピューターでできる繰り返し作業を人間がやる必要なんか全然ないじゃん、プログラムにやれる

ことはプログラムに任せて、俺達はもっと面白いことしようぜっていうのが、怠惰という考え方です。繰り返すのが本当に嫌い、単純作業嫌いということです。

それから「傲慢」、これは応用が利かないプログラムに腹が立つ、どういう意味かという、要するにその先を見越したプログラムを書こうじゃないかということです。結局プログラムを、これは先程の繰り返しの作業が嫌いというのと絡むのですが、応用が利かないプログラムを書いちゃうと、また次にちょっと変更があった時にまた書かなきゃいけないのです。プログラムはめちゃくちゃ面倒くさいことです。それをやりたくないから色々応用が利くプログラムを書いておけば、あとで楽できるんじゃないのということです。

最後は「人に悪く言われないプログラムを書き、人に悪く言われないよう制御する」、ということかという、美しいプログラムを目指す、どんどん公開するっていう意味なのです。我々プログラマーは、美しいプログラムという言い方をよくするのですが、それが何かということの説明が難しいのですが、よく書いてスキのない無駄のないプログラムのことです。美しいプログラムを書いて、どんどんそれを公開してみんなで共有することで、よりよい世の中を目指しているのです。とにかく繰り返しが嫌いで、美しいプログラムを書きたくて、拡張性に富んでいるプログラムを書きたい、こういう人達がいるからプログラムっていうのは進歩しているのだよということをラリー・ウォールという人は言っています。プログラマーのこういう性格を前提にしますと、サービスのことが分かってくるとと思います。一番重要なのは「怠惰」という考え方で、同じことを繰り返したくない、この考え方はサービスの進歩に関わっているのではないかと考えています。

どういう意味かっていうと、楽をするため

には努力を惜しまない、何か健康になるためなら死んでもいいみたいな、反語的な感じなのですけど、楽をするためにとにかくプログラムを書きたい、とにかくプログラム以外のことで面倒くさい思いをしたくないみたいなことを思って、日々プログラムを書くわけですが、その結果自動化という考え方が生まれます。手作業がとにかく嫌い、コンピューターにやらせることができることは、コンピューターにやらせてしまえ、そういうことで色々なことが自動的にできる方向に技術が進化してきたのではないかと感じております。それは何かというと、検索エンジンです。滝日さんの話にもありましたけども、最初に生まれてきた検索エンジンというのはディレクトリ型、いわゆる Yahoo! が始めたものだと思うのですけど、人間がいろんな自分がいいと思ったサイトをどんどん登録して行って、それを人間が分類して、来た人がそれを見て使う、こういう目的にはこういうサイトがあるのだというのが、すぐに見つけられるようにするというのが最初のディレクトリ型と言われる検索エンジンです。Google が有名ですが Google が始めて無いのですけれども、それがロボット型と言われるエンジンになりました。ウェブサイトっていうのはリンクが貼ってあって他のページに飛べるのが特徴ですけど、ロボットプログラムが自動的にリンクを辿って色々なページを巡回して、たくさんのページのデータを収集して、しかもそのページ自体がどのくらい意味のあるページなのかということや、それから検索に使ったキーワードとの関連があるかを自動的に検索して、自動的にユーザーが入力したキーワードに近いページが自動的に出てくるのです。あえて自動という言葉をとくさん使ったのですけど、そういう感じでとにかく人の手を介しなくても、インターネットにページ、サイトがたくさんできていけば勝手に検索できるページが勝手にできて行って、プログラムを

書くこと以外に全く人の手を介さなくても検索エンジンの仕組みができていくというのが検索エンジンの進化だと思います。こういうのも結局プログラムを書くこと以外したくないという考え方に基づいて、できてきたことなのではないかと思います。ディレクトリ型にはディレクトリ型の良さもあるのですけれども、ロボット型のほうが有効であるということが分かってきたので、今はロボット型が主流になって、Yahoo! もディレクトリ型からロボット型のエンジンに変える動きになっています。

それからブログとか、あと CMS というのはコンテンツ・マネージメント・システムです。コンテンツを導入すると適当な形でウェブサイトにして、表示してくれるシステム、ブログもその一部なのですけれども、ブログは書く内容がコンテンツで、ブラウザで入力さえすれば勝手にウェブサイトができるというのがあります。ウィキって言われている、ウィキウィキとかウィキウィキウェブとか言われていますけれども、これもコンテンツ・マネージメント・システム、CMS の一部なのです。日記とかとは違ってキーワード単位でページを自動で、自分で作れる特殊なブラウザ上で全部のページの管理ができるウェブシステムがあります。こういったものは HTML でウェブサイトができていますが、いちいち HTML を書くのは面倒くさい、オレが書きたいのは内容であって、そんなページの構成とかはどうでもいい、いつも同じ部分は自動的にやってくれよという気持ちがあった人達から生まれたのではないかと思います。結果として HTML とかを知らなくてもウェブサイトページが作れるということになったので、大ブレイクしました。最初からこう誰でも使えるウェブにしようという気持ちもあったと思うのですけれども、それとは別にいつも同じページ書いて、1 ページ書き換えたら他のページも書き換えなきゃいけ



ないとかは本当に面倒くさい、だからそんなの全部プログラムでできるのだから自動でやればいいじゃんって思ったこともすごくあるのではないかと思います。RSSはコンテンツを自動取得しやすいようなデータ形式の作成で、ブログの内容といったものをXMLという形式で、配信するってことが非常にブームになったのです。なんでこういうものができてきたかという、コンテンツを自動的に取ってきて加工できるというニーズがあったからです。ウェブページからデータを取ってきて意味解析するという技術も提供されていますけれども、それはそれで結構大変なので、それだったら最初からコンピューターが分かりやすい形でデータ制御してあげれば楽じゃないということでも生まれたのがRSSです。先ほどテクノラティとか紹介されていましたが、ああいったものもRSSを背後で使っている部分もあります。

こういった新しいサービスが出てきて、面白いことになっているのですが、最初のきっかけとしては自動化したいということです。更にRSSオートディスカバリといって、ウェブサイトのXMLの中に、ページのRSSとかURLを埋め込む仕様というのがあるのですが、そういったのもウェブサイトにアクセスしさえすればRSSの場所が分かる、自動的に分かりたいという気持ちから生まれたものじゃないかと思います。

ウェブAPIというのはウェブの仕組みを使ってデータをプログラム同士でやりとりするという仕組みなのですが、いろんなことに使われています。データのやりとりをウェブの仕組みを使って自動化していこうという動きです。自動的に最新データを取得したり、自動的にデータを更新したり、コンピューター同士でのこういったデータのやりとりというのはたくさんあるのですが、重要なのはウェブの仕組みを使ったということです。これは汎用的なプロトコルで、汎用的

な仕組みなのです。ウェブの仕組みってというのはどんなプログラミング言語でもたいてい使うことができる仕組みなのですが、こういうのを使うことでデータのやりとりなんかも簡単に出来る、あんまりプログラム書かなくていい自動化ができるというのが、これの仕組みのポイントだと思います。

次は広告についてです。最近アドセンスとってウェブサイトにもGoogleが出す広告を貼り付けていっているページがあります。検索結果に近いデータの内容の広告が出るので、クリックされる率も高いし、広告効果も高いし、しかも関係ない広告が出るのに比べると、見に来た人にとっても負担が少ないということで、よくできた仕組みです。Googleはアドワーズの収益が、かなりを占めていると言われていまして、Googleの世紀の発明とかという感じもしますが、これも広告だって自動的に表示したい、どういう風に広告出すのとか、広告主と相談したりとか、そんなことするの本当面倒くさい、だったら自動的に表示されるようにすれば、何も悩むことないし、アドセンスにしても、決まったHTMLの単語書いておけば、あとは勝手にGoogleがよさげな広告出してくれるので、広告出される側というか、ウェブサイトの管理者は何も考えなくていい、なんて便利、自動化素晴らしい、プログラムでできることはどんどんやりましょうということで、結構自動化されてきた部分があると思います。便利だったのでブレイクしましたけれども、自動化ということが根底にあるのです。

色んなことが自動化によって動いてきたと思うのですが、自動化できないこともあるということを話したいと思います。例えば、コンテンツの意味的分類というのは、結局自然言語処理です。日本語とか英語とか書いてあって、どういう意味を持っているかというのを解釈する研究はすごくされていて、昔に比べたらよくできていますが、やっぱ

りまだ実用段階には至ってないと思うのです。そのホームページ上にある、ウェブサイト上にあるデータというのを意味的に解釈しなくちゃいけない、最近は音楽ですとか動画ですとかといった、文字ですらないものがいっぱい出てきて、そういったものをどういう風に解釈してくのという時に、まだ技術的に難しい、今そういうのを、自動的に解釈してということはできません。じゃあどうする、みんなでやってたかって作業しなきゃいけないじゃん、という話になります。1人でやるのはとにかく面倒くさいので、1人でずっと続けていく根気は飽きっぽいプログラマーにはないのです。そこでフォークソノミーとか集合知という考え方、つまりたくさんの人が、ウェブ、インターネットに繋がっているから、みんなで作業したら、何か1人はみんなのためにみたいな感じになって、楽じゃないっていうのが、こういったもののきっかけの1つにあるのではないかなと思います。他にいろんなメリットというのも発見されてきました。情報に対してカテゴリ分類をするのではなくて、この情報に対するいろんな人が好きな言葉でキーワードを割り振っていくタギングと言われているものがあります。例えば、猫の写真があった時に、「猫」と付ける人もいれば「cat」と付ける人もいたりして、若干ブレはあるんですけど、カテゴリ分類では分類できないようなあいまいさみたいなところも、全体としてはキーワード表現できたりして、便利な分類方法ではないかと言われている分類方法です。これはどんなタグがあって、どのくらい使われているかというのを文字の大きさで示すタグクラウドって呼ばれるものです。タグをみんなで付けることでデータに意味が持たされてくる、みんなでやってたかって作業しようと、サービスのマネージャーですとかそういう人達が1人でシコシコ作業しなくても、みんなにこういう分類を開放するからみんなでやってくれよというこ

とで、情報が結果として分類されてしまうという便利な方法です。

次は、コンシューマー・ジェネレイテッド・メディア、CGMについてです。これもプログラム書いておけば、あとは消費者がどんどん内容を生成する、ユーザーが全部中身作ってくれるのではないかという考え方に基づいている部分が、ブログと同じようにあります。みんなが参加してくれることで楽しいとか、1人では絶対無理なことが出来るというメリットをもたらしたのです。スタート地点に「こういうサイトやりたいけど、中身作るの面倒くせー。みんなが中身を作れるようにしたら、みんなが作ってくれるんじゃないかなー」とを考えた人が絶対いると思っています。こういったこともプログラマーが、ギークが怠惰だったということがすごく関係していると思っています。

この間 Google が出したサービスなのですが、イメージラベラーという面白いゲームがあります。どういうゲームかというと、Google が画像をインターネットから拾ってきて画像をポンと出すのです。すると、自分とインターネットで繋いでどこかに仲間がもう1人、パートナーがいて、お互いにその画像の意味に近い言葉、単語をどんどん打ち込んでいくのです。パートナーと自分が同じ単語を打ち込んだら点数がもらえて、次の画像に進むというゲームです。同じ画像に対して、自分とパートナーがいかに同じ言葉を入力できるかということを考えるゲームなのですが、実はこれタグ付けじゃん、ということで、これを使って Google はどうもイメージにタグ付けをしているらしいという、巧妙なプログラマーの罠が潜んでいるわけです。これはよくできていて、その怠惰さというのは、聞こえは悪いのですが、そういった考え方によっていろんなものが分類されるようになってきているという、おもしろい例だと思います。

つまり、Google はギークの怠惰っぷりがウェブの進化の原動力になっていると考えているわけです。極端なことを言えば、淘汰の結果、理屈は後から付いてくる、ギークが色んなことをやって、色んなことを試した結果いいものだけが生き残ってきていると言えるのではないかと、ギークが作ったサービスの陰には死屍累々で、おもしろかったものが生き残っていると言えるのです。自然的にいいものが生き残ってきて、それはなんで良かったのかな、なんで生き残ったのかを考えてくと、色んな理屈が見えてくるということがあるのではないかと思います。

最近オープンなデータが入手出来やすくなっています。データが重用という話をしたいと思います。ウェブ API というのはウェブの仕組みを利用してデータと情報のやりとりをする仕組みのことです。他のウェブサイト上にある情報を XML でプログラマーが加工しやすい形で取得したりできるわけです。例えばアマゾンで、商品情報はウェブ上で検索すればたくさん出てくるわけです。その情報は、人が見やすいように HTML で整形されていて、その中から価格はどうで、作者はどうで、出版社はどうでというもうプログラムで抜き出せるけれど、どこにあるかという情報がないから面倒です。XML の場合は情報に対して意味付けをする仕組みができあがっているので、「この情報は作者です」「この情報が出版社です」「これが価格です」と埋め込んだデータを返すことができます。そういったことをすることで、様々な情報がウェブから取れるようになるので、データを取得したりするという、API が盛んです。他にも API の仕事というのはいっぱいあり、ブログ API は、ブログの統合をブラウザ上でやるだけではなくて、データが XML のやりとりできるというのもあるのです。YouTube でも動画の情報を検索してデータを XML で取ってくるメディアが公開されています。色々な

たくさんの API が出てきて、それが面白い効果を発揮しているということを言いたいと思います。ウェブ API には、特定の処理を実行するものと、サービスを操作するものと、データを取得するものの3種類があるので、XML でデータを渡して何かの XML のデータ返してくることが、なんでもできるわけです。例えば、あるブログの内容を書き換えるなら、「この内容書き換えて」「OK」みたいなこともできるし、「この計算やってみて」「OK」と単純に計算するだけのものもできるし、「この情報ちょうだいよ」「じゃあこの情報あげます」とやりとりもできます。重要なのはデータを取得する API 流行ってきたことです。色んなサイトからデータを取ってきて、加工や合成をすることで、新しい価値を生み出すことができるようになって、マッシュアップと言われています。例えば、グーグルマップ API という Google の地図情報を取得できる API があります。地図を表示できるグーグルマップという地図のサービスがあるのですが、それを使ってその上に自分で自由な情報を出すというようなシステムを作ることができる API です。地震が起きた時に地震の情報を取ってくる API を繋ぎ合わせて、地震が起きた場所を地図上に表示するサービスを作ってみたり、音楽情報とアマゾンの商品情報を合わせて、この音楽を聞いている人はこんな本買ったらいいいんじゃないと紹介するサービスとか、いろいろあるのです。とにかくいろんなデータをいろんなところが最近公開してくれるようになったことで、ほぼ無料でみんなが情報を加工した新しい情報を出せるようになってきています。ウェブ API が、サービスを外部プログラムから利用可能にしたことで、情報の共有や自社サービスの拡大、第三者による新しい視点の導入ができますというように、100 点満点の目的を達成した例なのではないかなと思っているのですが、これは結果として見えてきたことでしかないの

はないでしょうか。

2006、今年7月にオレゴンのポートランドでOS コンベンションがありまして、実は僕も行ってきたのですが、ここでオライリーは、最近重要なことが5つあるというような話をしています。これからはオープンソースのプログラムがうんぬんという話じゃなくて、「Open Source Licenses are Obsolete」と書いていますけども、オープンなデータ、そのデータをどう扱うか、オペレーションがアドバンテージだみたいなことを言って、データが色々公開されてきたことが将来に意味があることなんだということを強く言っていました。これは重要なことで、いろんなデータがほぼ無料で一般に公開されるようになってきている現状が重要なことだと思います。

なんでそのデータが重要なのかというと、オープンなデータはギークにとってかなり恰好のオモチャなのです。学研が昔出していた電子ブロック、色んなブロックを組み合わせていくとパルスを発信する何かになってみたり、お風呂の検知システムになってみたりと、いろんな電子工作ができるというもののなのです。これと同じで、オープンなデータというのは、オモチャのブロックのように組み合わせることでいろんなことができるのです。そういうことをするのが、ギークは大好きなのです。好奇心旺盛で新し物好きなので、新しいデータとか新しい物が出てくると、これを使って何かしてやろうと思うわけです。オープンなデータがいっぱい出てくることで、いろんな可能性の模索ができるようになってきたことが、新しいサービスが生まれる重要なポイントになってきているのではないかなと思っています。アマゾンとか YouTube とか、グーグルマップとか Flickr とか、API を公開するサービスに、ギークが群がっているようなサービスものを作り始めるのです。例えば、Flickr の画像を使った神経衰弱、グーグルマップの地図を使ったレースゲームとかある

のですが、そういったゲームみたいなものから、もっと役立つものまで、API を公開していると、それだけで他の人がそれを使って色々作ってくれるので、サービス自体が盛り上がっていくという傾向があります。API を公開しているかどうかというのは、そのサービスが流行るかどうかの分岐点だったりするのではないかなと思っています。

API 公開したからといって、その API を使うかどうかというのはギークにとっては微妙な選択肢になっていて、API 公開するサービスだから流行るということではないのではないかなと思います。どういう API を使いたいのかという、まず面白いデータが取得出来る、これ使ったら何かおもしろいことができるんじゃないの、と思うような何かが取れるものが重要です。それから標準的な正しい仕様になっていることです。例えば XML も 1 つの仕様ですが、いろんな仕様とかルールが今まで経緯から生まれてきています。そういったものに、きちんと則っているかどうかというのが使ってくれるかどうかの分岐点になっています。変なオレ流フォーマットでデータを出してきても、何だかワケ分からないし、コード書いても汎用性がないし、拡張性もないからあまり使いたくないと思われるとうと使われない、だから標準的な仕様というのをきっちり守って、その上で面白いデータを公開する API というのは流行りやすいと思われます。「この API のデータを使っていいけどこれで商売しちゃいけません」とか「このデータ使っていいけど第三者に公開しちゃいけません」とか「1日に1000回しか使えません」とか、いろいろと制限をかける場合がありますけど、ギークな人達はそういうのは嫌いです。「Information Wants To Be Free」という、「情報は自由になりたがる」という、超有名な編集者のスチュアート・ブランド 1984 年に言った言葉があります。これ自体は情報って隠すことができな

いんだよという意味でちょっと違うんですけど、やっぱり制限をかけられるっていうことはやっぱり嫌いですよね、何かこれっていずれこの制限が強くなったりするんじゃないかとかですね、いきなり使っちゃダメとか言うんじゃないかとか色んなことを考えちゃうので、制限がかかってないものを使うっていうのはこのところにかかっているかなと思います。事例として、日本のテレビ局がYouTubeに動画の事前審査を求めていることについて、先日日経新聞に出てたのですが、YouTubeに違法動画がいっぱいあがっていて、それ自体は問題を孕んでいたりとかするので、いいか悪いかは明言しないのですが、YouTubeに、日本のテレビ局が動画公開する前にチェックしろよというお願い文書みたいなものを出したのです。コンテンツホルダーからしてみれば、そういう考え方もあるのかも知れませんが、YouTubeもちっちゃな会社で、ギーク達が作っていると思うのです。事前審査は自動化が難しいし、面倒くさい、しかも自由な感じじゃないということで、ちょっとギーク的に考えれば「ねーよな、これ」っていう判断になるかなと思っています。もちろん、今後さまざまな外圧によって、もしかしたら事前審査が入るかも知れません。分からないですけども、そのプログラマー達は絶対嫌だと思っているという風に僕は思っています。その理由というのが先程言った怠惰さで、自由じゃない感じが嫌だとなるかなと思います。

結論なのですけれども、ギークは怠惰で、面倒くさがり、しかも新しい物、楽しい物が好きだといえます。ウェブサービスの将来はギーク“も”握っている。ギーク“は”って言えないところがちょっと弱いのですが、ギークのことを知ることも結構関係あるんじゃないかなという風に感じているということを伝えたいと思っています。

最後に、ちょっとオマケです。もし、自分でサービスを作ったことがない方は、自分でちょっとしたサービスを作ってみたりすると、ギーク達の生活や気持ちが分かったり、将来が見えてきたりするのではないかなと思っています。別の言い方をすると、コードを書けということで、今回の私の話は終わりのです。

うちの会社の日常をちょっとだけお見せします。こんな感じで仕事しています。非常に子供っぽい、大学の研究室みたいな感じの会社なのです。バラバラな机でごちゃごちゃした中で仕事をしているのです。開発者が多い会社なので、コードを重要視している人が多いので「社長がコードを書かなくなったらみんな言うこと聞かなくなるんじゃないですか」という発言が出たり、僕が最近、打ち合わせとかが多くなってきたら、「最近コード書いてないみたいですけど大丈夫ですか」って、いろんなことが言われていています。

ギークの人達っていうのは理論とか理屈では物事を考えないので、何かプログラムした結果のものを見せないと、納得したりしないのです。こういうのが面白いと思うのですけどみたいな、机上の空論みたいなプレゼンをする、「ふーん、そうかそうか。じゃあモノを作ったら考える」みたいな感じになっちゃって、とにかくモノを作らないとダメなのです。こういうことをよく言われて、コードが重要だということを再認識させられます。しかもわがままなので、やりたかったことも「上から言われると萎える」とか言って、あまり上の人言うことを聞かないギークな人達には多くて、好き勝手やっているのですけれども、そういったことの中から面白いものが生まれてくるのではないかなという部分もあるという風に思っています。それが全ていいかどうかは分からないのですけど、このようにやっています。

ご静聴ありがとうございました。

司会(高田)：ありがとうございます。それでは、質問などありましたらお願いします。

千葉 このスライドの最後のところに「ギークは正しい」というのがあるのですが、正しいという内容について少し聞きたいんです。

水野：正しいというのはかなり広い概念なのですが、ここで言う正しいというのは、美しいとかかなり似ている概念で、例えばちょっと前に流行ったポリティカリ・コレクトというものがあって、政治的に正しいとかいう言い方がありますよね。あれと結構近いかなと思うのですが、要するに自分達が進むべき方向というのがある程度、そんなに先ではないけど見えてくるわけなのです。例えば今後データというのはプログラムにとって解釈しやすい方向に行くとかです。さっきも今後流行るものという話が出ていましたけど、例えばマイクロフォーマット、ページの中にいろいろなもののデータを埋め込んでいこうという動きが流行ってきているとか。あと最近、そうだ、1つ思いついたのが、カレンダーサービスとかいうのが今後もっとくるだろうなと思っているのですが。そうやって自分達が今後面白いと思って進んでいく未来がそんなに遠くは見えなくても近くは見えているのです。それに対してその方向にある考え方とか、情報とかというのも正しいと見ていて、それと反対の方向を向いて、例えばYouTubeに対して事前審査が入るみたいな話では、結局どんどん自動化していこうという流れの中では、後ろ向きなわけですよ。そういうものは正しくないのではないかなと。いろいろな意味での正しさの中で、ギーク達が今後興味を持って進めていったりする方向に限定した上で持っていること。事前審査というのが政治的に正しいのかも知れないけど、ギーク的には正しくないということです。

千葉：ギークの方々の正しさのルールของเซตは共有されているのですか。

水野：結構ギークの人達というのは、意外とブログとかで喋ったり、いろいろなことを書いたりとかで、それなりに情報を共有する仕組みができ上がってきているので、先程のブックマークの仕組みですとか、RSSを使ったりとかといったことで、結構データの共有というのはできていて、割と何がおもしろいかというのは、共有されています。とんがった、みんなが同じような色をしている部分というのもあるけど、それが問題かなとは思っているのですが、データの共有みたいなことはできていると思います。

千葉：我々ギークでない者が正しいということと、ギークの方が正しいということとで、一致していない、矛盾しているというのがありますか。

水野：一致していないといったことよりも、理解してもらえないということは結構あるのです。最近見ていて思うのは、楽天というECサイトがありますよね。いっぱいサイトがあっという間にいろいろなもの売っていますが、デザインを見ていると、ずいぶん昔風なページもいっぱいあって、色がすごいとか文字がでかいとかいろいろあるのですが、そういうものの方が実は何か売れているような気もしていて、一般の人がウェブサイトとかを見て、ここはいいとかここはよく分かるといったことと、我々と全然見方が違っているなとは感じます。データを加工しやすいようにして、コンテンツとデザインの分離とかという話をよくするのですが、そういった形のことは我々は正しいと思っていますけど、そんなことよりも、派手な色を使って売れるサイト作った方が、商売してる人には正しいよなって思うこともあります。そういっ

たことにギャップを感じます。

**小池：**ギークと呼ばれる人達の自分のモットーとして、普段オレはギークだと思いながらプログラムを作ったり、オレは怠惰と思いながら何かやっているということはあるのでしょうか。それとも周りから見て、ああこの人はギークだなということなのか、自分もオレはギークだと、だからこういう怠惰でいいんだと、そう思いながらやっているものなのでしょうか。

**水野：**そこまで強く普段意識することはあまりないです。自分達がエンジニアであるとは思いますが、怠惰であるっていうのは、プログラムを書いていると結構自然にそうなうような気がして、「プログラムでできるんだからプログラムでやればいいじゃん」って思う人達が一定数出てきて、結果としてその人達がギークと呼ばれてる感じがします。ギークって言葉自体は、「オレ達はギークなんぞ」という感じでちょっとギャグっぽく使うことはありますが、まじめに「僕等はギークですから」みたいなことをいう人はあまり聞かないです。

**小池：**多分、怠惰という言葉は Perl の作者が出したっていうのが大きくて、怠惰っていうのは我々が普段使う怠惰とはかなり違うと思うのですよ。正しいっていうのも多分そうだと思うのです。怠惰というのは楽するためには労を惜しまないっていうような、実は裏の意味があって、繰り返しをしないためにコードを書くっていうのはかなり努力して、新しい技術を覚えているわけです。そういう背景の中で Perl の作者の影響を受けてひと括りで怠惰って言っているという感じもあるので、同じ怠惰なんだけども意味が違う風に言っているのかなっていう感じがしたんです。

**水野：**基本的にプログラマーが怠惰っていう時の、面倒くさいことをやりたくないっていう中には、プログラミングとプログラミングの勉強は入ってないですね。プログラミングとプログラミングの勉強はいくらしてそれはすごく楽しいことなので、快感なわけです。その状態をちょっと冗談めかして怠惰と呼んでいる感じではあると思います。

**小池：**プログラミング好きな人、そういう人だったら仕事とも思っていないので、好きなことをやっている、周りから見ていると苦痛だと思うのですが、働いている彼等にとっては苦痛でも何でもない、多分そういうことなのでしょうね。

**水野：**プログラミングを、基本的に1日会社でやっていて、家に帰っても、さあプログラミングしようっていう人も、うちの会社にいっぱいいるので、そういう人はまさしくおっしゃる通りだと思います。

**藤本：**ギークの類義語でハッカーがあったじゃないですか。ハッカーは結構馴染みがあるのですが、ギークはやっぱりよその世界っていう感があるのです。エリック・レイモンドが「ハッカーになろう」の中で「コードを書けないやつはハッカーじゃない」と明確に言って、ハッカーっぽくなりたくてやたら色んなサイトに入って侵入するとか、子供じみたことやつはハッカーじゃないからねみたいを書いていて、偽ハッカーと本物のハッカーを分けました。何故偽ハッカーが出るかというと、ハッカーってカッコイイじゃんみたいな、ちょっとワルになってみたい、プログラムがちょっとできるような子供達もしくは青少年、その人達がカッコイイという思いがあったと思うのです。ギークという響きの中にはそういうのはあるのですか。つまり、「オレはギークだ」と言いたいけど、でもコー

ドあんまり書けない偽ギークみたいなのがいるのか。

**水野：**それっぽい人には会ったことはあります。ギークな人達は本当にプログラミング大好きで、僕も好きなのですが僕は他にも好きなことが多すぎて、僕自身もギークではないと思っているのです。勤め先の取締役に「ウェブ進化論」を書いた梅田望夫というのがいるのですが、彼に、僕は最近ギークのことがだいぶん分かってきましたという話をしたのです。そしたら「で、君はギークじゃないの？」って言われたのです。僕は「どうも、ギークのことが分かれば分かるほど自分がギークじゃない気がして」って言ったら、「ああそうか、実は自分もそう思って、昔はプログラム書いてたけどそれでやめたんだ」という話をしたのです。ギークって、素質が強いなと感じているのです。なので、ギークになろうと思ってなれることはないのです。

**藤本：**ハッカーという言葉が、90年代後半インターネットが普及する過程で、侵入者とか悪者というようにメディアに使われて、元々プログラミングが大好きで、今あの水野さんが説明されたギークの特性を持った人達がハッカーという風に呼ばれて、元祖ハッカーというのは称号であったはずなのに、いつの間にか手垢にまみれてマイナスのイメージが普及されたっていう印象があるのです。人と何かハッカーのことを喋る時には常に何か言い訳をたくさんして、いい者と悪者もいるみたいなワケ分らない状況があったと思うのですが、水野さん達の世界では自分達をハッカーという定義はしないのですか。もしくはハッカーという言葉はもうないのですか。

**水野：**プログラミングが上手いっていうか天才的プログラマーみたいな意味で、ちゃんとハッカーという言葉を使っています。ハッ

カーよりも、ハッキン、ハックするっていう言葉がけっこう使われています。プログラムの一部におかしいところがあって直さなくちゃいけない時に、きれいに直そうとすると手を入れるところがいっぱいあって面倒くさいので、とりあえず今だけちょっと直しておこうということをクイックハックって言うのです。最近ハッカソンという言葉が生まれてきていて、マラソンみたいにずっとこうハックする、夜中じゅう一晩とか二晩とかずっとコード書くことをハッカソンという言葉があったりと、ハックとかハッキングという言葉も脈々と生き残ってはいます。死語ではないです。

**鈴木：**卒業生の鈴木と申します。ギークはプログラミングが好きで休日でもプログラミングをして、仕事とプライベートの区別がない、そこで例えば「はてな」のCTOの伊藤直也さんなんかは、早い段階からブログの検索システムを作って、そういうものが会社の中に還元されて会社のサービスが始まったりしていますけども、そういうことに対して会社から個人に対しての報酬のようなものはあるのでしょうか。

**水野：**ギークな人達含めてウェブ関係で働いているエンジニアは、得なきゃいけない知識があるので好きじゃないとやっていけない部分があるとは思っています。好きだから休日でも情報収集しているし、プログラムも書いているし、そういうのは業務でも生かせるとみなされる感じなので、よっぽどすごい何かを生み出したってことになれば話は別なのかも知れないですけども、知る限りでは無いのではないかと思います。

**鈴木：**それは逆に言うと技術に追われている立場で、常に新しいことを吸収しないと次の新しいサービスを生み出せないということが



あると思うのです。僕の印象では、ハッカーみたいな人達はC言語のような基盤技術をしっかりおさえているのという強みがある人達で、ギークっていうのはどっちかって言うウェブサービス中心に開発を進めている人達じゃないかっていう印象があるのですけど、新しいサービスを生み出さないといけないギークの人達はどういうところに行き着くのか知りたいなと思います。

若い人達を中心としてウェブサービスが作られてますけども、この先若い人達がこのまま同じことを続けていけるんだろうかということを知りたいのです。

**水野：**C言語が書けるか書けないかということに関して言えば、よくできる技術者でC言語を書いたことがない技術者もいます。Cとかが全然分からない感じなのですがですけども、やっぱり新しい技術を覚えるのが好きでいろいろなことをよく知っているのです。C言語が分からないということと、基盤の技術が分からないということは違うと思っていて、例えばメモリ管理がどうされているかってことは、ちゃんと知っているし、問題はないと思

います。ただC言語は分かっていたほうがいいよなみたいな話を、この間会社でしたら、「いやあ、誰でも自分が最初に習った言語が一番いいって言うんですよ」って言われたので、ああそうかもなと思っています。確かに学ぶことはどんどん増えていくので、全体を見通せる人というのは今後少なくなってきちゃうのかも知れないですね。その昔は発明家が1人で、例えばエジソンが電球作って、電話作ってとか、できたかも知れないけど、今はどんどん細分化されています。テレビのシャープの亀山工場でテレビの画面のガラスを精度高く作れるエンジニアもありなわけです。そういった形で、もしかしたら今後は細分化されて、今みたいな形のギークっていうのは消えていくのかも知れません。何か常に勉強することはあるし、将来は見通せない感じはするのですが、違った形で変わっていく可能性はあるのではないかなという風に思います。

**司会：**他になれば、終わりたいと思います。どうもありがとうございました。

**水野：**どうもありがとうございました。